Computer vision 1 - Lab 1

Daniel van de Pavert (11045418), Damiaan Reijnaers (10804137), Shuai Wang (13128051)

September 2020

Introduction

In this research report, several subjects are being treated: photometric stereo, colour spaces, intrinsic image decomposition and colour constancy.

The most important part of this research consists of a thorough outline of the *photometric stereo algorithm*, which can be summarized as an algorithm to 'backward engineer' the laws of math and physics to infer the depth of an object by passing in a (number of) input picture(s) with an object being illuminated from different angles.

In the colour spaces section we convert a RGB image to several different colour spaces and analyze the composition of the channels. Using the albedo and the shading properties of a ball we reconstruct the original image in the intrinsic image decomposition section, we also recolour the image. The final section will be on colour constancy where the grey world algorithm is used to colour balance an image.

1 Photometric Stereo

1.1 Estimating Albedo and Surface Normal

Albedo measure

From images given in *SphereGray5*, we can find that there are four parts made from two kinds of material. Since the albedo measure conveys the total amount of diffuse reflection—reflection of light scattered in all directions— of (solar) radiation (as a ratio of the total solar radiation), it can be inferred that an image graphing this value would also consist of four different parts for this particular set of images. Hence, our result, which is shown in figure 1, is as expected.

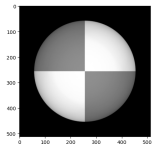


Figure 1: Albedo image

An important note has to been made on the type of objects shown in the images in the dataset – it is assumed they diffuse into a *Lambertian* reflectance pattern, as the current implementation of the photometric stereo algorithm limits its applicability to *Lambertian surfaces*[1].

Increasing performance

The photometric stereo algorithm bases itself on one important equation: the intensity of a pixel value as the result of a dot product between a vector modeling the ray of light falling on the object, and a reflected ray of light normal to the surface which is scaled by the albedo value (a measurement of the amount of light reflected). Since one 'part' of this vector dot product is unknown, we have to solve a linear system of equations. Since the dot product is performed on vectors of three dimensions, at least three equations are needed. This means that we either need three gray-coloured photos (since photos consisting of grayscale values have only one *channel*[2] or at least one coloured photo, since a typical coloured photo consists of three channels: *red*, *green* and *blue* (Drew, 1992).

[1] A Lambertian surface is a surface whose brightness, as a result of light reflection, appears the same regardless of viewing angle.

[2] Images usually consist of a combination of the primary colours *red*, *green* and *blue*, the 'pixel intensities' of an image for each of these primary colours, or a single gray-scale colour, is called a channel.

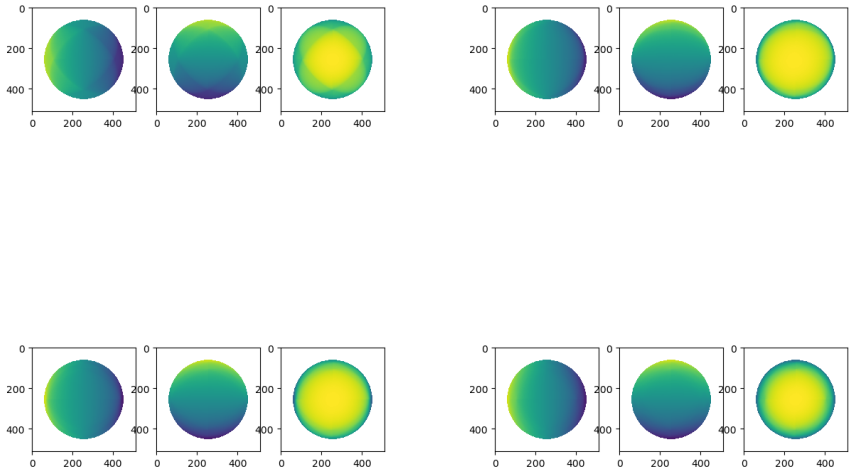


Figure 2: Surface normal, sampled size of 5 (upper left), 10 (ur), 15 (bottom left) and 25 (br)

As shown in figure 2, the precision with which we can reproduce the surface normal vectors increases as the number of sample images grows larger.

Shadows

The presence of shadows is one of the major problems of the photometric stereo algorithm (Hernández et al., 2010, p. 1). By multiplying both sides of the equation of the pixel intensity value involving the dot product, as mentioned in the previous subsection, with a diagonal matrix of these same intensity values, 'shadowy regions' (with either low intensity values, or *zero*) are canceled out – a simple 'trick' partly solves the problem. If we would have kept these 'shadow points,' the mentioned dot product (of the ray of light and the surface normal vector) in these regions would equate to 0, which is clearly wrong. For our implementation, applying the 'shadowing trick,' unexpectedly, performs worse than not applying the trick, as shown in figures 3 and 4.

As a diagonal matrix of intensity values for a point corresponding to all given mages is made for every pixel in the image dimensions, storing this whole 'shadow-trick matrix' (denoted by scriptI in the exercise) might be computationally useful, but infeasible if working with a large number of input pictures. As modern graphics cards are optimized for operations on matrix, we have tested both an approach including as much matrix operations as possible (in which, for example, np.einsum('ijkl,ijk-¿ijk',scriptI, i) could calculate the matrix/tensor multiplication operation for the left-hand side of the mentioned equation. We have implemented a hybrid application of both methods.

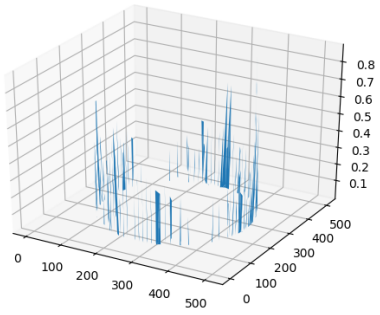
1.2 Test of Integrability

There are a number of reasons why the 'sanity test of integration' could fail: shadows, light direction errors and, among others, image red intensity errors. About the test with reading different images, for reading 5 photos, Number of outliers is 8425, for reading 10 photos, number of outliers is 5699, for reading 15 photos, number of outliers is 4185,for 20 photos:3075; for 25 photos 2337. Hence, with the increasing of photos numbers, results are more and more precise.

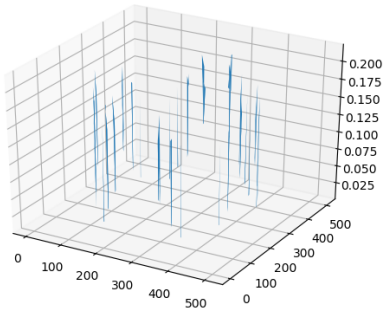
1.3 Shape by Integration

Integration from different 'paths'

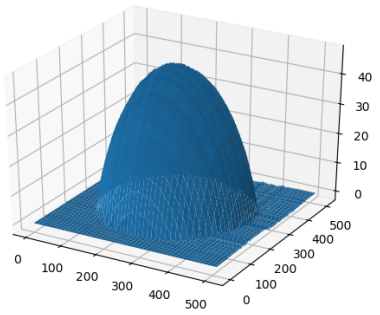
The photometric stereo algorithm cleverly makes use of the fact that the *gradient* of a surface is always *normal* to that same surface. As already indicated, in figure 5, by plotting these normal



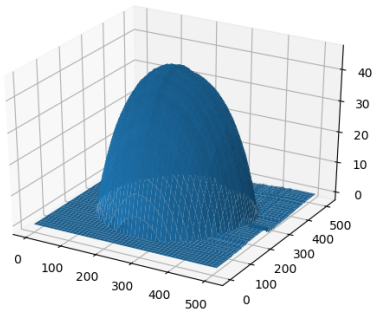
(a) Outliers, trick applied



(b) Outliers, trick not applied

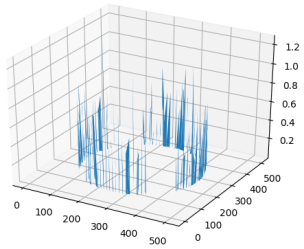


(c) Graph, trick applied

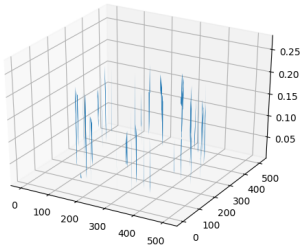


(d) Graph, trick not applied

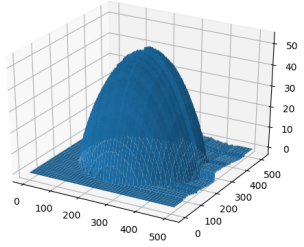
Figure 3: Effects of 'shadowing trick' on SphereGray25



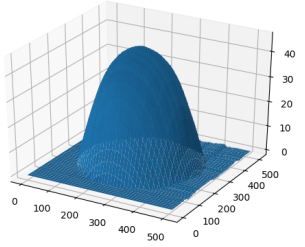
(a) Outliers, trick applied



(b) Outliers, trick not applied



(c) Graph, trick applied



(d) Graph, trick not applied

Figure 4: Effects of 'shadowing trick' on SphereGray5

vectors to the surface, it is possible to reconstruct the surface itself by making 'backward use' of the notion of the gradient. As the gradient of a function f of a vector $\boldsymbol{x} \in R^n$ is denoted as $\nabla_{\boldsymbol{f}} = \begin{bmatrix} \frac{\partial f}{\partial x_1} & \cdots & \frac{\partial f}{\partial x_n} \end{bmatrix}$, we can simply write our derived normal vector *in terms of* the gradient of a function at a point on the surface – the function we need to reconstruct the surface itself. Since the data structure of our digital image is in two dimensions (the width and height of a photo), while the 'real life object' is in three dimensions, we have a function $f\colon \mathbf{R}^2 \to \mathbf{R}^3$ that maps a value $f(x, y)$ to a *depth*-value of the object. If setting p equal to f, our derived gradient consists of $\begin{bmatrix} \frac{\partial p}{\partial x} & \frac{\partial p}{\partial y} & \frac{\partial p}{\partial z} \end{bmatrix}$, and thus, $\frac{\partial f}{\partial x} = \frac{\frac{\partial p}{\partial x}}{\frac{\partial p}{\partial f}} = \frac{a(x,y)}{c(x,y)}$ for $f(a(x, y), b(x, y), c(x, y))$. This way, we only need to keep track of the x and y direction of the gradient, and we can store these, for the dimensions of pixels in the given pictures, in separate matrices p and q.

Since the gradient in a certain point on the surface—the actual surface of the object that we would like to reconstruct—can be intuitively captured as being 'small steps in the direction of the surface of steepest increase,' we can reconstruct the object's surface solely by using the gradient's small increments in every point. If we start at a point in euclidean space (most preferably, $(0, 0)$) we can reconstruct the object's surface by creating a *depth map* by cumulatively summing over all directional gradient values in each point. That way, we can either 'walk' across all pixels in one direction (for example, across *one* column, or *one* row) and then loop over *all but one* pixel (the one we already did in the previous 'walk') in the orthogonal direction. Note that walking over rows is equivalent to the transpose of the depth map resulting from walking over the columns with transposed p and q.

Both in the row- and column direction, minor discrepancies are visible – mainly in the bottom surface of the object. This might be the result of shadows and irregularities in the given images, and its negative effects can be reduced by averaging the two depth maps resulting from walking in both directions, as illustrated in figure 6.

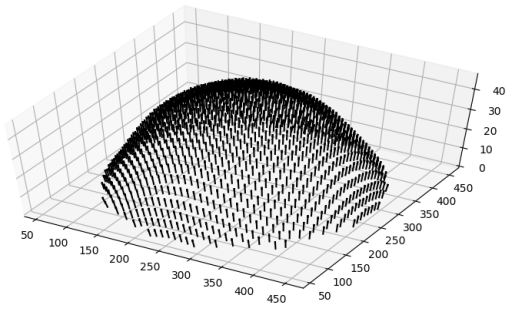


Figure 5: Quiver plot of normal vectors derived from 5 gray-scale images of a spheric object

1.4 Experiments with different objects

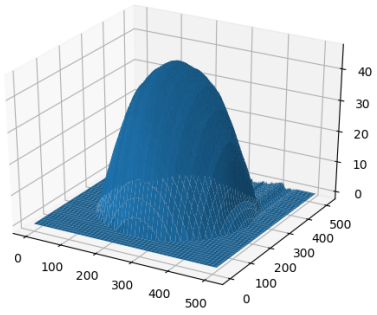
A monkey in gray

Reasons for creating for errors maybe the light resources are incorrect due to object's complicated surface

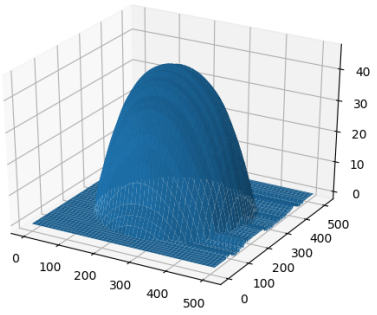
A monkey in colour

By treating each of the *red*, *green* and *blue* channels independently, we end up with $3 \cdot n$ sample pictures for n input pictures. This results in more data, which results in slightly lesser number of outliers: **7496** in case of the monkey, as opposed to **8059**. The use of colour seems to significantly improve the depth graph of the sphere, as seen in figure 6d.

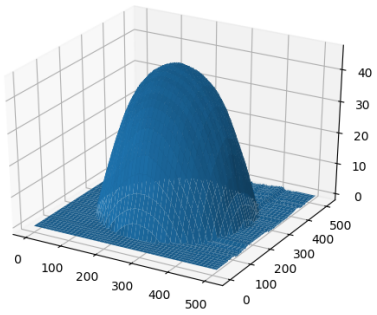
If a certain channel (for example, the *blue* channel) amounts of only pixels of zero value, this will cause similar errors as for which we implemented the 'shadowing trick' in section 1.1. To overcome this problem, we can discard such 'empty channels' entirely.



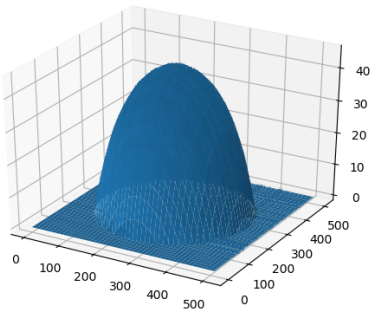
(a) Reconstruction over columns



(b) Reconstruction over rows



(c) Averaged row/column reconstruction



(d) Subfigure c for 25 coloured images

Figure 6: Surface reconstructions for 5 gray-scale images of a sphere

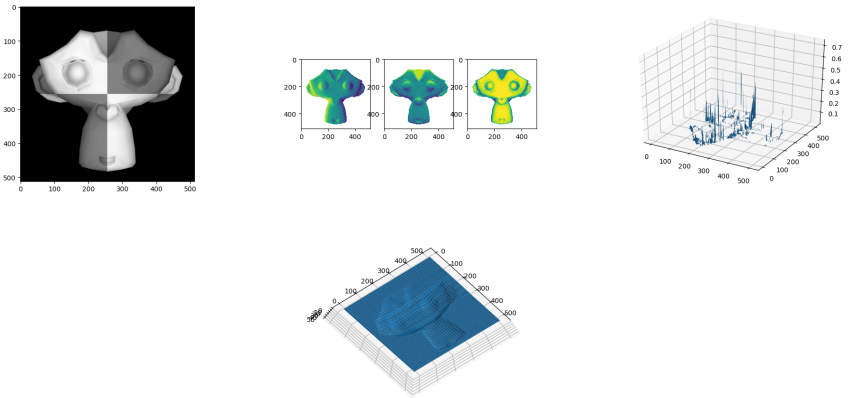
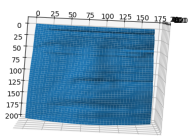


Figure 7: Albedo, Normal, SE, Surface of Monkey

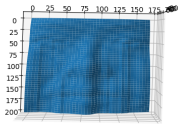
A more complicated way to implement RGB-channeling is to introduce a *channel constant* for every channel in the input dimensions – this constant can then be derived after back-substituting for the independently derived normal vectors[3].

Yale faces

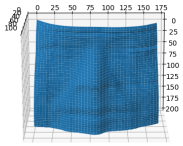
As shown in figure 8, different path algorithms yield different result in the case of inputting close-up images of faces. This might be the result of human faces not being entirely symmetric, as was the case in all of the other example objects encountered in this research lab.



(a) Reconstruction over columns



(b) Reconstruction over rows



(c) Averaged row/column reconstruction

Figure 8: Surface reconstructions for Yale faces

2 Colour Spaces

2.1 RGB Colour Model

Digital cameras use a set of three cells per pixel to compute the colour. These cells are all attuned to a different portion of the colour spectrum named for their primary colour. Any spectral colour can be represented as a linear combination of these primary colours. This additive property is convenient in cameras and the reason most mammals have RGB cones.

2.2 Colour Space Properties and Conversion

2.2.1 Opponent Colour Space

In Herring's Opponent Process Colour Theory, described in Bratkova et al. (2009), Herring proposes that the observable colour space is not divided in three sections as the trichromatic theory of Young-Helmholtz but in four sections. These sections are in addition to the trichromatic primary colours (Red, Green, Blue) Herring adds the colour yellow as it does not seem to be create-able using only red, green and blue. It is commonly used in color transfer applications.

$$\begin{pmatrix} O_1 \\ O_2 \\ O_3 \end{pmatrix} = \begin{pmatrix} \frac{R-G}{\sqrt{2}} \\ \frac{R+G-2B}{\sqrt{6}} \\ \frac{R+G+B}{\sqrt{6}} \end{pmatrix}$$

[3]Paul G. Allen School of Computer Science Engineering, *Computer Vision, CSE P576, December 2005, Lab 5*, `https://courses.cs.washington.edu/courses/csep576/05wi/lectures/photostereo.ppt`, accessed on September 14th, 2020

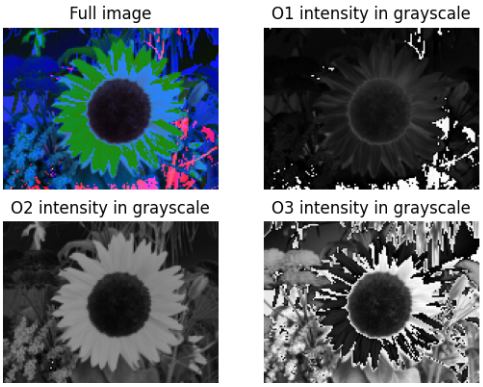


Figure 9: The opponent colour space, channels

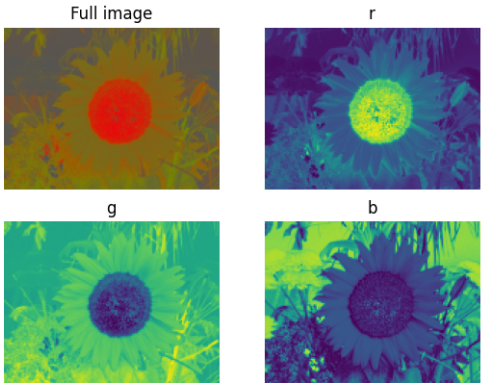


Figure 10: The normalized rgb colour space, channels

2.2.2 Normalized RGB (rgb) Colour Space

Normalizing the RGB colour space leads to a colour space independent of illumination and surface orientation. Using this manner of representing the colour spectrum a colour gamut can be formed using any three colours (points) on the plane. With these points any colour inside the gamut can then be formed by mixing the chosen colours. There can however be seen that there is no such gamut that can include the whole visible spectrum. This colour space is widly used in white balance algorithms.

$$\begin{pmatrix} r \\ g \\ b \end{pmatrix} = \begin{pmatrix} \frac{R}{R+G+B} \\ \frac{G}{R+G+B} \\ \frac{B}{R+G+B} \end{pmatrix}$$

2.2.3 HSV Colour Space

The HSV colour space has a separate channel for the image intensity and colour information where in the RGB colour space these are mixed. The separation of these channels is convenient when doing things like histogram equalization as the colour component should most likely be left alone but the intensity component should be worked on.

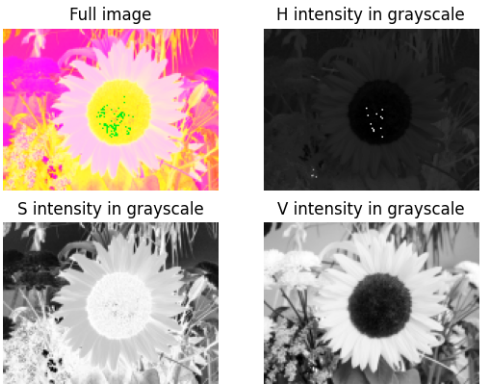


Figure 11: The HSV colour space, channels

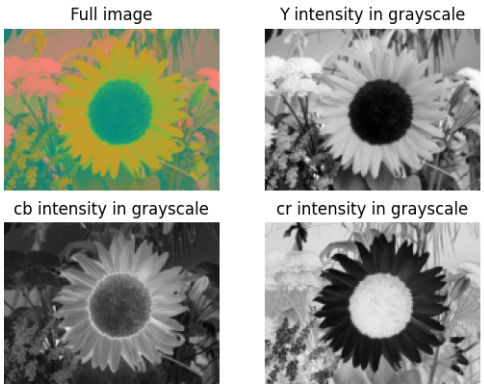


Figure 12: The YCbCr colour space, channels

2.2.4 YCbCr Colour Space

As described in Podpora et al. (2014) the YCbCr colour space, a version of the YUV colour space provides a better subjective quality in comparison with RGB images when tested on subjects. (Podpora et al., 2014) conclude that the use of the YUV colour space is better for machine vision implementations than RGB as the YUV colour space has perceptual similarities to the human vision.

2.2.5 Grayscale

Grayscale image tend to be easier to work with as they are comprised of far less data and focus more on the intensity of the light. These features can help with colour invariant tasks like edge detection.

Figure 13: Grayscale - Average method

Figure 14: Grayscale - CV2 method

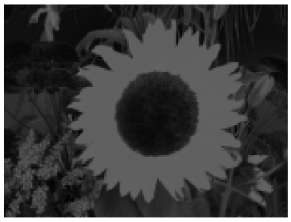


Figure 15: Grayscale - Lightness method

Figure 16: Grayscale - Luminosity method

2.3 More on Colour Spaces

As described in Ibraheem et al. (2012) the CMY colour space is based on complementary colours (Black, Cyan, Magenta and Yellow) and primarily used in output systems like printers. In contrast to the additive nature of the RGB colour space the CMY colour space is subtractive meaning colours are formed subtracting one or more primary colours from white.

3 Intrinsic Image Decomposition

3.1 Other Intrinsic Components

As seen in Starck et al. (2003) images can also be decomposed into texture and piece-wise smooth content. This method is also used in image analysis and image synthesis.

3.2 Synthetic Images

Fixed controllable light sources and surfaces with a singular reflective value are easier to create in a synthetic image. These values can be created in a lab environment but the creation of data sets would be a far greater amount of work in comparison to rendering the images with a computer.

3.3 Image Formation

Figure 17: Original ball

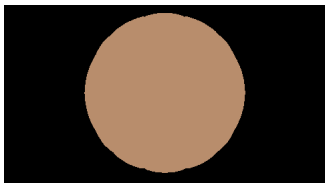


Figure 18: Albedo of the ball

Figure 19: Shading of the ball

Figure 20: Reconstructed image of the ball

3.4 Recoloring

The true material colour of the ball is: R: 184, G: 141, B: 108.
Though the ball recoloring with green colors, final images are not totally green due to shading RGB, which reduces the intensity of the colors. The shading component is not uniform, and thus the colors of the resulting images are not uniform either.

Figure 21: Ball

Figure 22: recoloring ball

4 Colour Constancy

1. After the Grey-World algorithm, we can see that the picture is more nature and looks better.
2. The gray world algorithm might fail in images without a good variation of colours. In these kinds of images the average RGB colour might be quite far from grey. This could happen when a white light is shone upon a room filled with primarily red objects, the average RGB value would not be grey despite the lighting conditions being optimal. This would lead to a colour shift that leaves the image looking faded or unnatural.
3. The white patch algorithm assumes that somewhere in the image a patch of white colour is available and scales the colours to that maximum. A more specific variation of this approach is the scale-by-max approach where the the maximum in the image is found by looking for the maximum value in each channel. This algorithm is inefficient when no good white patch can be found in the image.

Conclusion

In question 1 we learned the how to implement Photometric Stereo, which contains calculate the Albedo Normal vector of the face and finally reconstruct the surface from single to multi-color.In the colour spaces section we saw the applications of different colour systems and the relative ease of switching between them. In our research we did see that there are various methods of transforming one colour system into another but we found the transformations given in the assignment to be best

for this application. The intrinsic image decomposition showed how an image can be reconstructed using its components and using those same components recoloured. With the colour constancy section we concluded that the grey world algorithm can work quite well even though it is one of the simpler algorithms for colour correction. We also briefly looked into the white patch algorithm, another simple colour correction algorithm that would sufficiently work on our test image but also has its flaws.

References

Bratkova, M., Boulos, S., and Shirley, P. (2009). Orgb: A practical opponent color space for computer graphics. *IEEE Comput. Graph. Appl.*, 29(1):42–55.

Drew, M. S. (1992). *Shape from color*. Simon Fraser University, Centre for Systems Science.

Hernández, C., Vogiatzis, G., and Cipolla, R. (2010). Overcoming shadows in 3-source photometric stereo. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 33(2):419–426.

Ibraheem, N. A., Hasan, M. M., Khan, R. Z., and Mishra, P. K. (2012). Understanding color models: a review.

Podpora, M., Korbas, G. P., and Kawala-Janik, A. (2014). Yuv vs rgb-choosing a color space for human-machine interaction. In *FedCSIS (Position Papers)*, pages 29–34.

Starck, J.-L., Elad, M., and Donoho, D. L. (2003). Image decomposition: Separation of texture from piecewise smooth content. In *Wavelets: Applications in Signal and Image Processing X*, volume 5207, pages 571–582. International Society for Optics and Photonics.